CV HW3

tags: `Computer Vision`

Group 23

- 0856005 Liu, An-Chi
- 0853426 Chen, Shao-Yun
- 0856123 Ko, Ruo-Lin

Introduction

Image stitching is the process combining several images with overlapping fields and produce a panoramic image. Also, the logical flow bwtween the images must be preserved.

There are many image stitching algorithms exist. The most commonly used are direct approach and feature-based approach. The direct approach use all the image pixels to compute the correlation functions as a result of highly computational expensive.

In this assignment, we will perform feature-based technique. We use SIFT or SURF as feature detector and use those features to find the relations between different images.

Implementation procedure

1. Load both images, and convert them to double and to grayscale.

2. Detect feature points in both images by SIFT.

```
siftDetector= cv2.xfeatures2d.SIFT_create()
kp, des = siftDetector.detectAndCompute(img, None)
```

3. To find the matches between the key points of the left image and the right image, we use the brute-force matcher, which takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation, and the closest one is returned. The process is as the following:

    1. We extract local neighborhoods around every key point in both images, and form descriptors simply by “flattening” the pixel values in each neighborhood to one-dimensional vectors.
    2. Compute euclidean distances between every descriptor in one image and every descriptor in the other image.
    3. Select putative matches based on the matrix of pairwise descriptor distances obtained above. We select all pairs whose descriptor distances are below a specified threshold.

4. For a matched-points pair, We need to find the homography H for the given point P of the left image and the given point $P'$of the right image, which satisfies the follow equation:

$$w\mathrm{P}' = \mathrm{HP}$$

    where w is an arbitrary coefficient of H.

5. In order to solve for homographies, we extract the above equation to:

$$\begin{bmatrix} wx'_i \\ wy'_i \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

which is equal to:

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i y_i & -y'_i \end{bmatrix} = \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathrm{A} = \mathrm{h0}$$

There are many pairs of matched-points, so we have a $2n \times 9$ matrix A.

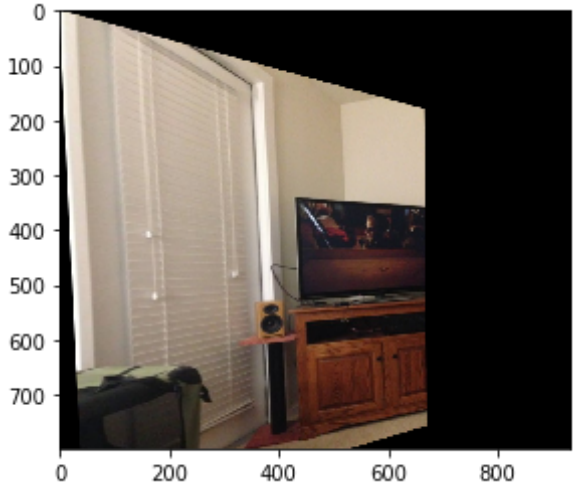
The solution of the h is well-known as the eigenvector of $\mathrm{A}^T\mathrm{A}$ associated with the smallest eigenvalue.

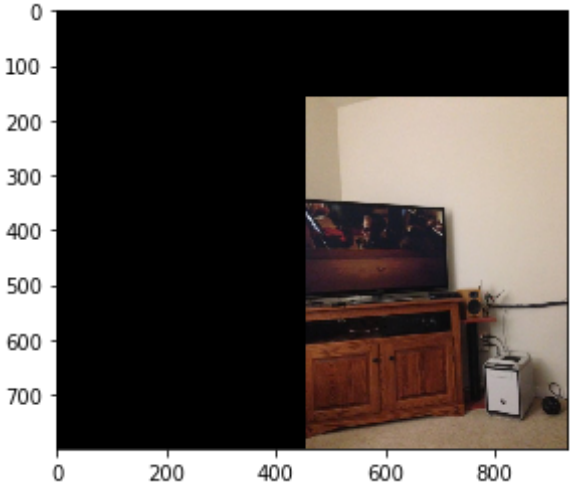
6. Since we just need 4 pairt to get a homography H, we can get a lot of Hs. Therefor the RANSAC algorithm is adopted to estimate the best homography mapping one image onto the other.

7. Random sample Consensus (RANSAC) executes the following steps:

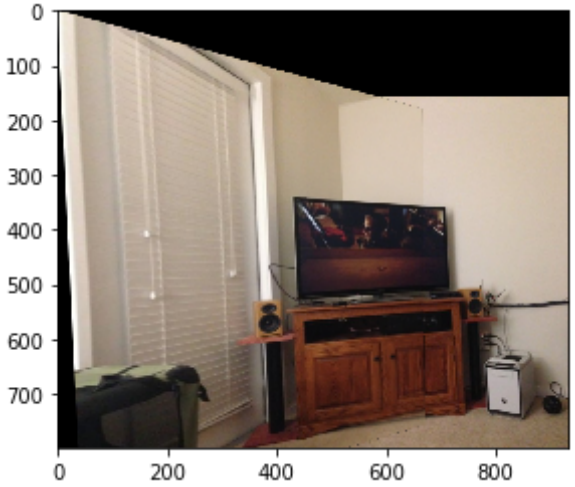
    1. Select randomly the minimum number of points required to determine the homography H.
    2. Solve for the homography H.
    3. Determine how many points from the set of all points fit with a predefined tolerance.
    4. If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ , re-estimate the model parameters using all the identified inliers and terminate.
    5. Otherwise, repeat steps 1 through 4 (maximum of N times).

8. Warp both images onto the other using the estimated transformation(H).





9. Create a new image frame, which is big enough to hold the panorama and composite the two images into it.



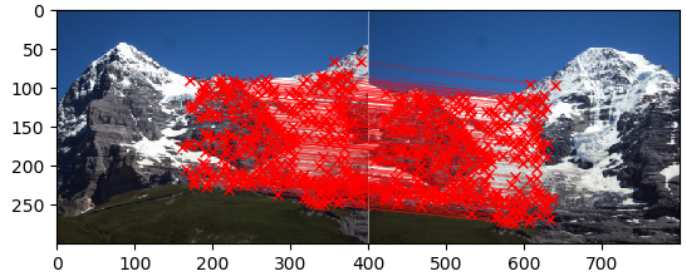
Experimental Results

Images from HW3 data

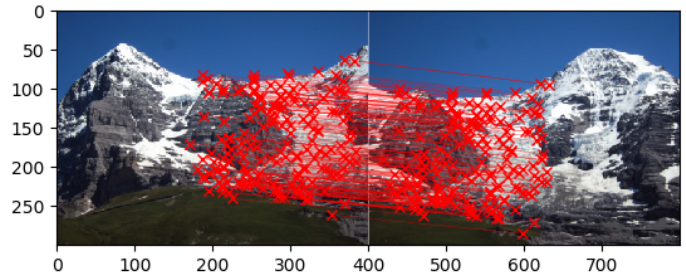
Left image

Right image

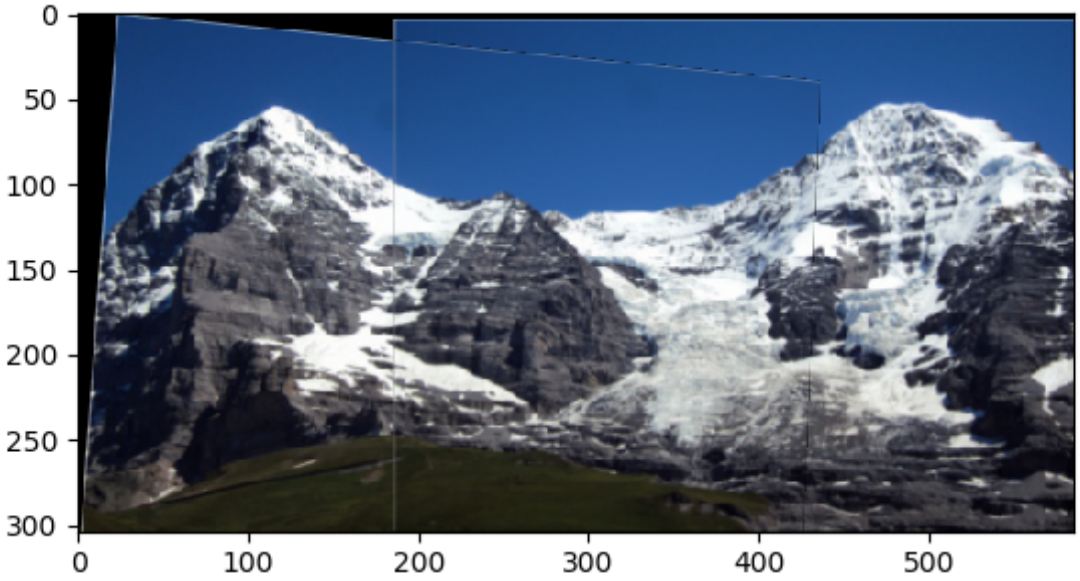
Feature matching by SIFT



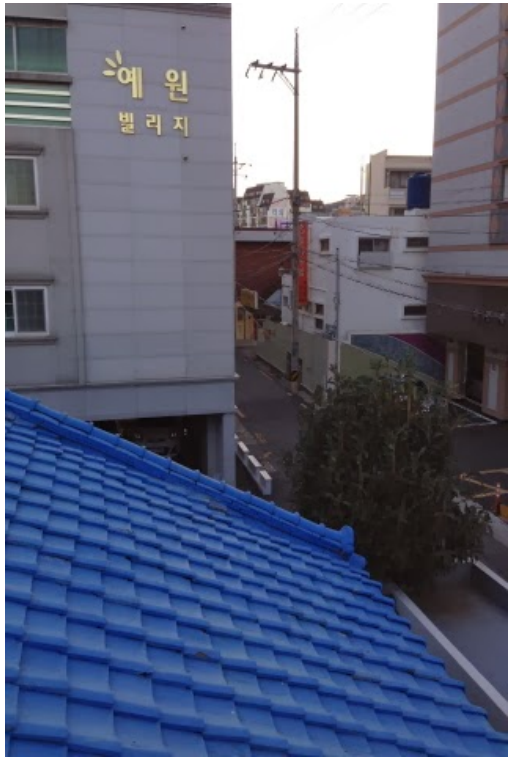
Feature matching by SURF



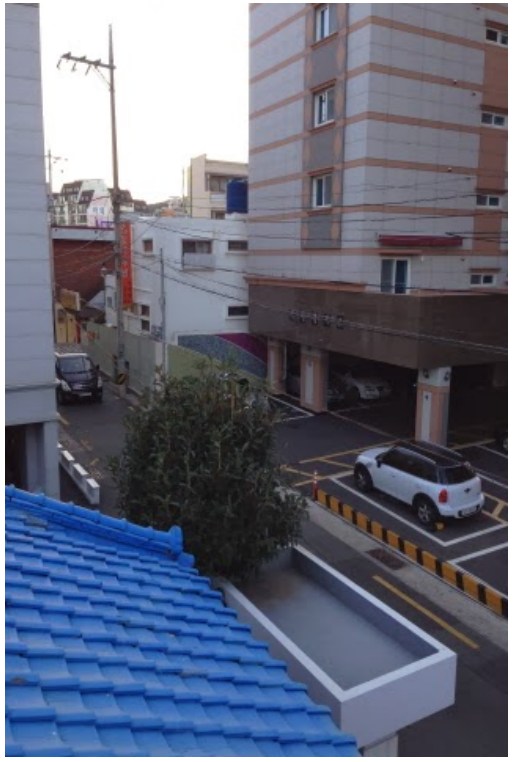
Panoramic view(Result), same as using SIFT or SURF



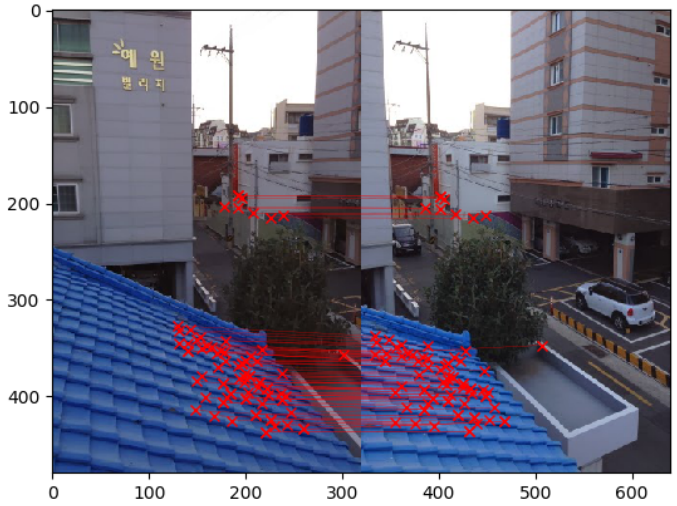
Left image



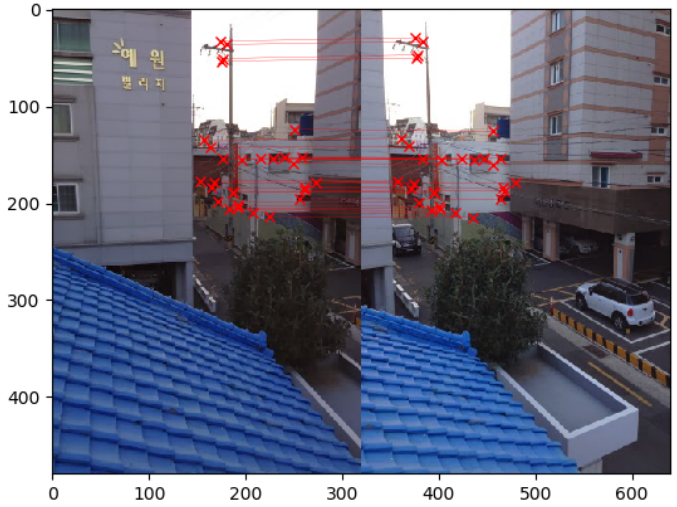
Right image



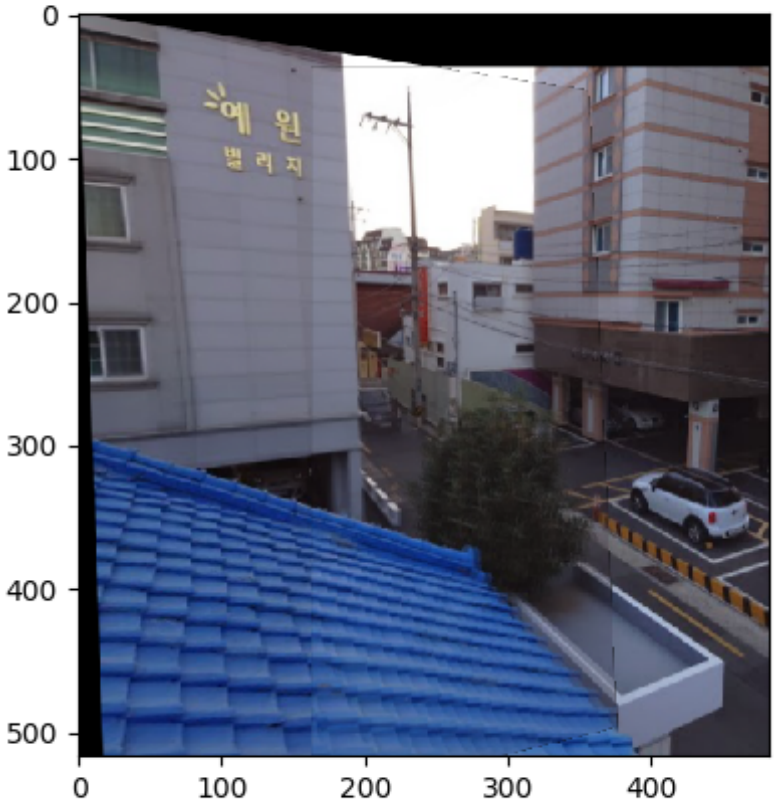
Feature matching by SIFT



Feature matching by SURF

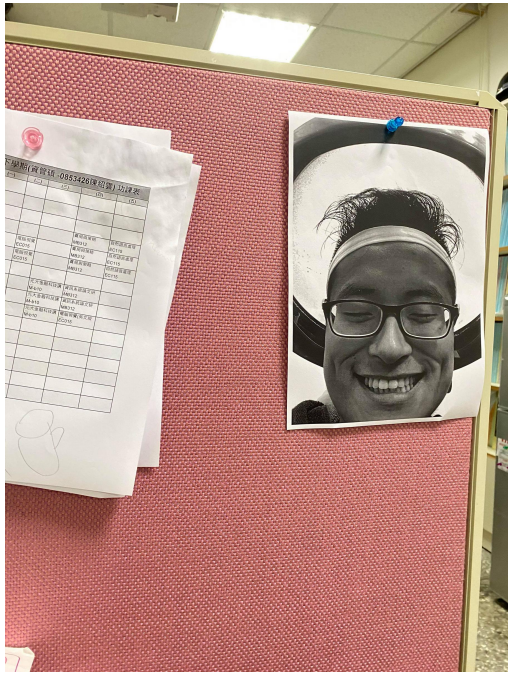


Panoramic view(Result), same as using SIFT or SURF



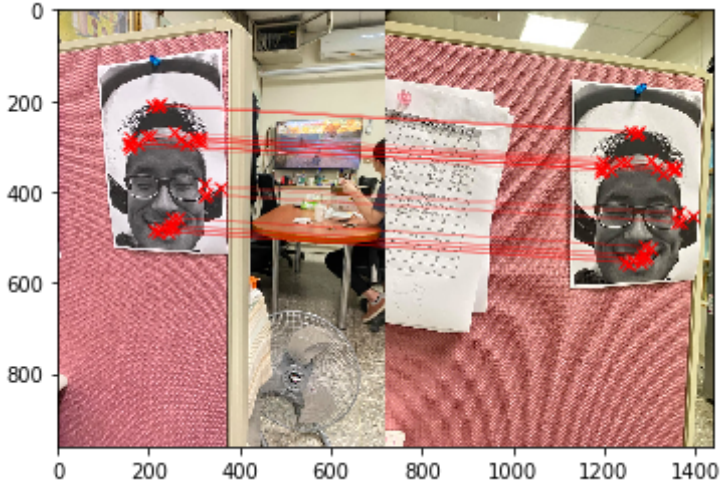
Our own images

Left image



Right image

Feature matching by SIFT



Panoramic view(Result)

Discussion (what difficulties you have met? how you resolve them?)

1. We use cv2 to read the images and convert them to gray. However, we didn't notice that cv2 use BGR channel. It caused we did't successfully convert the images to RGB and GRAY. This error will cause the SIFT detection result. Consequently running RANSAC algorithm would't get correct result (a few inliers).

2. `cv2.xfeatures2d.SIFT_create` is deprecated in the latest OpenCV, so it is required to intalled a 3.4.x version.

3. The values of descriptor obtained by SURF are very small. We need to select the threshold value of `def match_points(kp_left, kp_right, des_left, des_right, threshold=0.01)` very carefully or it may be either no match point or the match points are incorrect of two images.

Conclusion

In this assignment, we try two different feature detectors to find interest points and feature descriptions. The result panoramic images are the same using these two detectors.

Work Assignment Plan

- Liu, An-Chi: implementation procedure
- Chen, Shao-Yun: Implementation procedure, Experimental Results, Discussion
- Ko, Ruo-Lin: Introduction, Discussion